ABSTRACT

In many continuous speech recognition systems based on HMMs, decision tree-based state tying has been used for not only improving the robustness and accuracy of context dependent acoustic modeling but also synthesizing unseen models. To construct the phonetic decision tree, standard method has used just single Gaussian triphone models to cluster states. The coarse clusters generated using just single Gaussian models can lead to low accuracy acoustic modeling and result in low recognition performance of the system. In this paper, a multi-stage decision tree using both multi-mixture Gaussian models and single Gaussian models is proposed. Continuous speech recognition experiment using this approach on WSJ data showed a reduction in word error rate comparing to the standard decision tree based system.

1. INTRODUCTION

Mixture Gaussian distributions and context dependent phone models have been used to achieve high performance in many continuous speech recognition systems based on continuous density Hidden Markov Models (HMMs)[1]. In this case, data insufficiency problem occurs owing to the increased number of free parameters. Furthermore, the distribution of phones and contexts is usually uneven in general speech and speech database. This fact requires a method to balance between model complexity and available training data[2].

Smoothing[3,4], backing-off[5] and parameter sharing[6] are representative approaches to deal with these problems. Smoothing uses interpolation between the less and more detailed models. In this approach, it is possible that different parts of a model can be smoothed in variable proportions with different models. Backing-off is to throw away infrequent triphones and train only the most frequent triphones. The missing triphone is substituted with a biphone or monophone model. But the big jump in model specificity can introduce problems in accuracy of modeling. The third approach, parameter sharing is sensible because the acoustic realizations of a phone occurring in different contexts are often very similar[7,8].

There are two different parameter sharing schemes, model-based sharing(generalized triphone) and state-based sharing(state tying), according to sharing level. Model-based sharing has limitation in that the left and right contexts cannot be treated independently. Hence this may not be the most appropriate approach[2]. State tying can build more accurate Markov models by having left and right context modeled separately. The state-tying can be implemented using either bottom-up or top-down approach. To treat the problem of unseen triphone, which is more serious when using cross word context dependencies, top-down clustering based on decision trees has been used popularly[9]. Besides being able to synthesize unseen triphones, this approach has additional advantages when using phonetic decision tree. Firstly, expert knowledge can be incorporated in the form of phonetic questions for clustering. Secondly, through the construction procedure of decision tree, it is possible to ensure all models have sufficient examples in the training data.

Traditional methods[2,8] of construction of phonetic tree are using just single Gaussian triphone models to cluster states. Based on this clustered states, parameters are re-estimated and the number of mixture components is increased. The coarse clusters generated using just single Gaussian models can lead to low accuracy acoustic modeling and result in low recognition performance of the system. In order to get higher quality of the state clusters for robust parameter estimation, multi-stage decision tree using both multi-mixture Gaussian models and single Gaussian models is proposed. In section 2, we review the traditional phonetic decision tree and then describe newly proposed approach, the multi-stage decision tree. In section 3, overall system architecture and some experimental results are presented.

2. MULTI-STAGE DECISION TREE

In this section, the traditional decision tree and newly proposed decision tree algorithms are described.

2.1 Traditional Decision Tree

Phonetic decision tree is a binary tree. Each node has a question regarding the phonetic context of the triphone. The examples of the phonetic questions are as follows.

Is Right Nasal (*-m *-n *-en *-ng *-em) ?
Is Left Liquid (l- e l- r- w- y- hh-)?

For each state of a phone, a tree is built to cluster the corresponding states of all the associated triphones. The shape of the tree and the questions of nodes are chosen to maximize the likelihood of the training data given the tied states, the states in same terminal node. The tied states share the same parameters trained using all the observations assigned to this state set. To construct a tree, states are grouped first into a root node. The log likelihood of the training data is calculated using single Gaussian distribution on the assumption that all of the states in the node are tied. The approximation of the log likelihood, \( L(S) \), is given by

\[
L(S) = -\frac{1}{2} (\log(|\Sigma_s|) + n(1 + \log(2\pi))) \sum_{r \in S} \sum_{f \in F} \gamma_s^f
\]

where \( S \) is a set of HMM states, \( n \) is the dimensionality of the data vector and \( \gamma_s^f \) which has been accumulated during Baum-Welch re-estimation is the probability of the frame \( o_i \) being generated by state \( s \). It is assumed the assignments of observations to states are not changed during the procedure of tree construction. The node is then split using the best phonetic question out of a question set which yields the maximum increase in log likelihood for the two child nodes. The question is assigned to the node, and the states in the node are distributed over the two child nodes according to the result of question evaluation. This procedure is repeated until the desired number of tree leaves is achieved. After this building process the resulting decision tree is used for training and recognition. In training stage, the states of single Gaussian triphone models are clustered, and then mixture splitting and parameter re-estimation are followed. This approach can cause problem in the quality of the state clusters owing to using just single Gaussian triphone models. The coarse state clusters can lead to low accurate acoustic modeling and be one of reasons for low recognition performance.

### 2.2 Multi-Stage Decision Tree

To improve the quality of state clusters and get more accurate acoustic modeling, a multi-stage decision tree is proposed. In this approach, decision tree for state-tying is constructed and updated using both multi-mixture and single Gaussian models. The first decision tree using single Gaussian uses the approximation of log likelihood which is same with the one used in the traditional decision tree. Re-estimation and mixture splitting are performed with the adequate amount of data. After the re-estimation and the mixture splitting, decision tree is built again using multi-mixture Gaussian models. A sequence of these processes for acoustic modeling is repeated until the number of mixture components reaches desired one or the system get the best performance. During building the decision tree using multi-mixture models, the approximation for the probability of each node must be calculated. The pooled parameters for the pool of multi-mixture states are computed first. Pooled variance, mean and total state occupancy is given by

\[
\gamma_j = \sum_{i=0}^{N} \gamma_{ij}, 1 \leq j \geq M, 1 \leq i \geq N, \quad (2)
\]

\[
\mu_j = \frac{\sum_{i=1}^{N} \gamma_{ij} \mu_i}{\gamma_j}, \quad (3)
\]

\[
\Sigma_j = \frac{\sum_{i=1}^{N} (\gamma_{ij} (\mu_i - \mu_j)^2 + \Sigma_{ij})}{\gamma_j}, \quad (4)
\]

where \( N \) is the number of states in the state pool, \( M \) the maximum number of mixture components of a state. \( \gamma_{ij}, \mu_i \) and \( \Sigma_{ij} \) are the probability of state occupation, mean and variance of \( i \)th Gaussian of \( j \)th state in the states pool. The approximation of the log likelihood of the data is computed efficiently under the assumption that the assignments of data to states are not altered during building decision trees and that the total likelihood of the data can be approximated using the following expression.

\[
L(S) = \sum_{o_i \in S} \sum_{i=1}^{M} \log \{ P(o_i;G_i) w_i \} \gamma_i. \quad (5)
\]

where \( S \) is a cluster of states, \( o_i \) is all data frames assigned to specific state \( s \), \( G_i \) is \( i \)th distribution of the mixture and \( w_i \) is weight for \( i \)th distribution of the mixture. Assuming Gaussian distributions, \( L(S) \) can be written as

\[
L(S) = \sum_{o_i \in S} \sum_{i=1}^{M} \left( -\frac{1}{2} (n \log(2\pi)) + \log(|\Sigma_i|) + \right. \\
\left. (o_i - \mu_i) \Sigma_i^{-1} (o_i - \mu_i) \right) + \log(w_i) \gamma_i \quad (6)
\]

where \( n \) is the dimensionality of feature vector. Applying parameter re-estimation formula to equation (6) gives

\[
L(S) = \sum_{i=1}^{M} \left( -\frac{1}{2} \sum_{j=1}^{n} \right. \\
\left. (1 + \log(2\pi)) + \log(\Sigma_{ij}) + \log w_i \right) \gamma_i \quad (7)
\]

where, \( \Sigma_{ij} \) is \( j \)th diagonal element of covariance \( \Sigma_{ij} \). \( \Sigma_{ij} \) is not calculated directly from the data but calculated from the means and variances of the states in the cluster. The procedure of node splitting for building decision tree is controlled by the above log likelihood approximation \( L(S) \). This is used again to merge similar nodes after
splitting. During this acoustic modeling, the number of leaves of the decision trees is decreased gradually while the number of mixture components increases.

3. EXPERIMENT

The performance of the proposed multi-stage decision tree based state tying was evaluated on the Wall Street Journal (WSJ) 5k task. The standard S184 training data set was used in this system. 12 MFCC, the normalized energy plus 1st and 2nd order time derivatives were used as feature vector. All triphone HMM models have three states and left-to-right topology. The bigram language models and cross-word triphone models were trained with WSJ data. The multi-stage decision tree based state tying was used to cluster the context dependent states and to synthesize unseen triphones. The phonetic question set used for the decision tree consists of 217 questions. The maximum number of mixtures for each state is 10. For this experiment, more than 20 CPUs were used. Construction of decision tree, training, language modeling and decoding were performed in parallel on different CPUs. This could reduce execution time by more than 10 times. For efficiency and extensibility, the masking based on phone indexes was used for phonetic question matching.

To compare the proposed multi-stage decision tree with the traditional decision tree, we generated about 4000 tied states using each decision tree. Table 1 lists the number of tied states according to the thresholds used in the multi-stage decision tree, especially two-stage decision tree. Table 1 shows the different effects of thresholds upon the number of tied states generated at the 1st stage and the 2nd stage. The relationship between the number of tied states and threshold at 1st stage is same with the one of traditional decision tree.

<table>
<thead>
<tr>
<th>Tied states(threshold) at 1st stage</th>
<th>Tied states(threshold) at 2nd stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>4518 (360)</td>
<td>2990 (1250)</td>
</tr>
<tr>
<td>5012 (310)</td>
<td>3211 (1150)</td>
</tr>
<tr>
<td>5625 (290)</td>
<td>3754 (1000)</td>
</tr>
<tr>
<td>6123 (240)</td>
<td>4108 (900)</td>
</tr>
<tr>
<td>6810 (210)</td>
<td>4406 (750)</td>
</tr>
<tr>
<td>7924 (170)</td>
<td>5014 (650)</td>
</tr>
</tbody>
</table>

Table 1. The number of tied states and thresholds at the 1st and 2nd stage of multi-stage decision tree.

The relationship between the number of tied states of 1st stage and the one of 2nd stage was also investigated. Table 2 shows the number of tied states of 1st stage and word error rate when fixing the number of tied states of 2nd stage at 4108. The ratio of the number of tied states of 1st stage to the one of 2nd stage was 3 to 2 was considered to be appropriate. The phonetic decision trees using 1 mixture, 2 mixture, 4 mixture and 8 mixture Gaussian components per state were constructed and evaluated. Multi-stage decision tree using 2 mixture models showed better performance than traditional decision tree, but we could not see further improvements with decision tree using 4 and 8 mixture models. The reason seems to come from using approximation to calculate score for splitting and merging tree nodes.

<table>
<thead>
<tr>
<th>The number of tied states at the 1st stage</th>
<th>Word error rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4518</td>
<td>9.9</td>
</tr>
<tr>
<td>5012</td>
<td>9.5</td>
</tr>
<tr>
<td>5625</td>
<td>9.3</td>
</tr>
<tr>
<td>6123</td>
<td>9.2</td>
</tr>
<tr>
<td>6810</td>
<td>9.2</td>
</tr>
<tr>
<td>7924</td>
<td>9.1</td>
</tr>
</tbody>
</table>

Table 2. The number of tied states at 1st stage and corresponding word error rate when fixing the number of tied states of 2nd stage at 4108

With the 4050 tied states in the traditional decision tree, 6123 and 4108 tied states at the 1-stage and 2-stage of multi-mixture decision tree, the word error rates on WSJ 5k task is listed in table 3.

<table>
<thead>
<tr>
<th>Tied states</th>
<th>Word error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>4050</td>
</tr>
<tr>
<td>2-stage(2 mix.)</td>
<td>4108</td>
</tr>
<tr>
<td>3-stage(4 mix.)</td>
<td>4150</td>
</tr>
<tr>
<td>4-stage(8 mix.)</td>
<td>3985</td>
</tr>
</tbody>
</table>

Table 3. Comparison of the performances of traditional decision tree vs. multi-stage decision tree.

As can be seen in table 3, the performance of multi-stage decision tree is better than that of traditional decision tree. And 2-stage decision tree using 2 mixture Gaussian models showed the best performance among multi-stage decision trees.

4. CONCLUSIONS

In this paper, a multi-stage decision tree to get more accurate acoustic model is proposed. The multi-stage decision tree for state-tying is built and updated using both multi-mixture Gaussian model and single models while traditional decision tree is built with just single Gaussian models. In the proposed approach, a sequence of acoustic modeling which includes re-estimation, mixture splitting and decision tree construction is repeated until the number of mixture components reaches desired one or the system get the best performance. When constructing the multi-stage decision tree, the approximation for the probability of each node is calculated based on the pooled parameters. The pooled
parameters for the pool of multi-mixture states are computed not using data directly but with the statistics of each state in the pool. This approach was evaluated on the WSJ 5k data. Multi-stage decision tree using multi-mixture models showed better performance than traditional decision tree. We could get the most improvements with two-stage decision tree using 2 mixture models. In the near future, we will develop better formula to be able to calculate the score of the cluster of multi-mixture states when building decision tree for state-tying.

REFERENCES